

**VŠB - Technická univerzita Ostrava**  
**Fakulta elektrotechniky a informatiky**  
**Katedra informatiky**

**Herní server sudoku**

**Sudoku Game Server**

**2010**

**Bc. Jaroslav Bližňák**

# Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne 6.5 2010

.....  
Bc. Jaroslav Bližňák

## **Poděkování**

Chtěl bych na tomto místě poděkovat Ing. Emílii Šeptákové, za odbornou pomoc, věcné připomínky, rady a vedení při tvorbě této diplomové práce. Také bych chtěl poděkovat své rodině a kamarádům za psychickou podporu a trpělivost při realizaci této práce.

## Abstrakt

Cílem této práce je vytvoření herního serveru sudoku pro zájemce o tuto hru. Server umožní generovat zadání sudoku různých obtížností. Součástí herního serveru budou soutěže, kterých se budou účastnit registrovaní uživatelé. V rámci soutěží budou evidovány jejich průběžné i celkové výsledky. Implementace bude provedena ve vývojovém prostředí Visual Studio 2008 v jazyce C#. Implementace bude otestována na základě požadavků herního systému. V závěru budou uvedeny možná rozšíření tohoto systému a shrnutí získaných zkušeností.

## Abstract

The aim of this diploma thesis is to create a game server of Sudoku for those who are interested in this game. The server will enable to generate Sudoku tasks of various difficulty levels. There will be some contests as a part of the game server in which the registered users can participate. Their running as well as final results are to be filed in terms of the contests. The implementation will be carried out in the development system Visual Studio 2008 in the C# language and tested on the basis of requirements of the game system. In conclusion, there are to be stated possible extensions of the system and summing-up of the gained experience.

**Klíčová slova:** sudoku, hráč, soutěž, hra, jazyk C#, MS SQL, ASP.NET, AJAX

**Keywords:** sudoku, player, match, game, the C# language, MS SQL, ASP.NET, AJAX

## Seznam použitých zkratek a symbolů

<b>Zkratka</b>	<b>Význam</b>
ERD	Entity Relation Diagram
DFD	Data Flow Diagram
SQL	Structured Query Language
ASP	Active Server Page
MS	Microsoft
FCL	Framework Class Library
CLR	Common Language Runtime
CTS	Common Type System
CLS	Common Language System
AJAX	Asynchronous JavaScript And XML
IIS	Internet Information Services

## OBSAH

<b>1 ÚVOD.....</b>	<b>1</b>
<b>2 ÚVOD DO PROBLEMATIKY.....</b>	<b>2</b>
2.1 SUDOKU .....	2
2.1.1 Historie .....	2
2.1.2 Základní pojmy .....	3
2.1.3 Koncepce a pravidla hry.....	5
2.1.4 Jednoznačné řešení hry .....	5
2.1.5 Celkový počet řešení.....	5
2.1.6 Další varianty sudoku.....	5
2.2 POŽADAVKY NA HERNÍ SYSTÉM .....	6
2.3 ZÁKLADNÍ ROZDĚLENÍ PROBLEMATIKY .....	7
<b>3 GENERÁTOR ÚLOH SUDOKU .....</b>	<b>8</b>
3.1 POUŽITÍ HOTOVÝCH ÚLOH SUDOKU .....	8
3.2 ODEBÍRÁNÍ Z MŘÍŽKY SUDOKU .....	8
3.2.1 Generátor mřížky sudoku.....	9
3.3 PŘIDÁVÁNÍ DO PRÁZDNÉHO POLE SUDOKU .....	9
3.4 METODY ŘEŠÍCÍ SUDOKU ÚLOHY .....	10
3.4.1 Metoda Backtracking.....	10
3.4.2 Rekurzivní algoritmus Brute-force .....	11
3.4.3 Techniky napodobující logické myšlení řešitele .....	12
<b>4 HERNÍ SYSTÉM .....</b>	<b>14</b>
4.1 ROLE V SYSTÉMU .....	14
4.1.1 Anonym.....	14
4.1.2 Hráč.....	15
4.1.3 Administrátor.....	15
4.2 HRÁČ .....	16
4.3 SOUTĚŽ .....	17
4.4 HRA.....	17
4.4.1 Bodování.....	18
4.5 TÝMY .....	18
4.6 ČÍSELNÍKY.....	19
4.7 PRŮBĚŽNÉ A ZÁVĚREČNÉ VÝSLEDKY .....	19
<b>5 ANALÝZA.....</b>	<b>20</b>
5.1 DATOVÁ ANALÝZA.....	20
5.1.1 Lineární zápis .....	20
5.1.2 E-R diagram .....	22
5.1.3 Datový slovník .....	23
5.2 FUNKČNÍ ANALÝZA .....	25
5.2.1 Diagramy datových toků (DFD) .....	25
5.2.2 Minispecifikace.....	27
<b>6 NÁVRH IMPLEMENTACE.....</b>	<b>29</b>
6.1 ARCHITEKTURA SYSTÉMU .....	29

6.2	MODEL AUTENTIZACE A AUTORIZACE UŽIVATELE .....	31
6.3	MODEL DATABÁZE .....	32
6.4	TRANSAKČNÍ ANALÝZA .....	33
6.5	SYSTÉMOVÝ NÁVRH .....	34
6.5.1	ASP.NET .....	34
6.5.2	C# .....	35
6.5.3	SQL Server .....	35
6.5.4	AJAX .....	36
<b>7</b>	<b>IMPLEMENTACE .....</b>	<b>37</b>
7.1	ÚLOŽNÉ PROCEDURY A FUNKCE .....	37
7.2	POUŽITÉ KOMPONENTY .....	40
7.2.1	CalendarExtender .....	40
7.2.2	TabContainer .....	41
7.2.3	GridView .....	42
7.3	HRACÍ POLE .....	43
7.4	OBRAZOVKA UŽIVATELSKÉHO ROZHRÁNÍ .....	44
<b>8</b>	<b>TESTOVÁNÍ .....</b>	<b>45</b>
8.1	TESTOVACÍ SCÉNÁŘ PŘIDÁNÍ NOVÉ SOUTĚŽE .....	45
8.2	TESTOVACÍ SCÉNÁŘ REGISTRACE NOVÉHO UŽIVATELE .....	46
<b>9</b>	<b>ZÁVĚR .....</b>	<b>47</b>
<b>10</b>	<b>LITERATURA .....</b>	<b>48</b>
<b>11</b>	<b>PŘÍLOHY DIPLOMOVÉ PRÁCE .....</b>	<b>49</b>
11.1	OBSAH PŘILOŽENÉHO CD-ROM .....	49

# 1 Úvod

Tato diplomová práce se věnuje problematice herního serveru sudoku. Cílem je vytvoření webového portálu pro zájemce o hru sudoku. Tento portál umožní generovat úlohy sudoku různých obtížností. Součástí herního serveru budou různé soutěže v sudoku, kterých se budou moci účastnit registrovaní hráči. V rámci hodnocení jednotlivých soutěží bude systém vypisovat různé průběžné i závěrečné výsledky týkající se úspěšnosti hráčů.

Úvodní kapitola je věnována základním informacím týkajících se hry sudoku. Je zde popsána její historie, základní pojmy, koncepce, pravidla hry a mnoho dalších pojmů, které jsou důležité pro pochopení dané problematiky. Tato kapitola popisuje také požadavky na budoucí herní systém a dále se zaměřuje na rozdělení problematiky této práce.

Problematika herního serveru sudoku je rozdělena do dvou kapitol. V první kapitole je popsán generátor her sudoku, který je možno rozdělit do tří částí, podle způsobu jakým jsou dané hry vytvářeny. Druhá kapitola popisuje problematiku herního systému a provádí specifikaci činností jednotlivých částí systému, které lépe přiblíží jeho funkcionalitu. Popis funkcionality zahrnují: uživatelské role, soutěže, hry, týmy, hráče, číselníky a různé průběžné nebo závěrečné výsledky tohoto systému.

Další kapitoly se věnují analýze systému, která je rozdělena do jednotlivých fází. Poté je uveden návrh, popis samotné implementace a testování tohoto systému. Je popsána architektura a také použité technologie v systému, u kterých je popsán základní princip funkčnosti a ukázky některých použitých komponent.

Závěr obsahuje zhodnocení, zda výsledek této práce odpovídá kladeným požadavkům a cílům. Zmíněno je také možné rozšíření tohoto systému o další moduly, zhodnocení a osobní přínos pro řešitele této diplomové práce.



## 2 Úvod do problematiky

Aby bylo možno popsat problematiku této práce, je nutné si předem vysvětlit některé základní pojmy, principy a rozdělení, které blíže představí funkčnost této aplikace.

### 2.1 Sudoku

Prvním cílem této práce bylo vytvořit generátor hry sudoku, která bude generovat sudoku zadání různých obtížností. V této kapitole budou popsány všechny důležité informace týkající se této hry.

#### 2.1.1 Historie

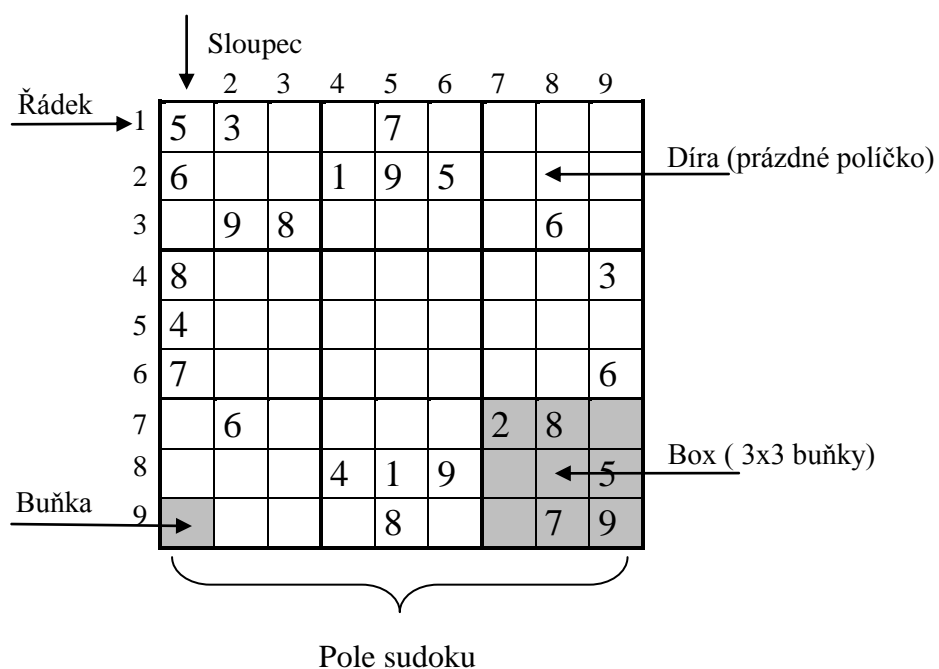
Hra sudoku je populární logická hra (hlavolam), která je v současné době hodně oblíbená u lidí všech věkových kategorií. Název hry vznikl z japonského „**Súdži** wa **dokušin** ni kagiru“, což může být přeloženo jako „čísla jsou omezena na jeden výskyt“ nebo „čísla musí být samostatná“. Z toho důvodu je hra většinou považována za japonskou, což není pravda [1].

Hra sudoku byla pravděpodobně navržena Howardem Garnsem, což byl 74 letý architekt z Indiany. Byl tvůrcem hlavolamů, rébusů a hádanek v penzi. Poprvé byla hra vydána v roce 1979 v americkém časopisu Del Magazines pod názvem „Number place“, což může být přeloženo jako „Umístí číslice“.

Hra byla v roce 1984 uvedena v Japonsku nakladatelstvím Nikoli pod názvem „**Súdži** wa **dokušin** ni kagiru“, které bylo později zkráceno na sudoku. V Japonsku se tato číselná křížovka stala rychle populární - možná také proto, že v Japonštině klasické slovní křížovky prakticky nelze tvořit. V roce 2004 přenesl sudoku do Anglie Wayne Gould. Ten propadl této hře natolik, že vytvořil program, který dokázal vygenerovat nespočetné množství variant sudoku a tento program nabídl zdarma londýnskému listu The Times. Od té doby začali docenovat její krásu a zdánlivou jednoduchost lidé ze západní civilizace včetně České republiky [2].

### 2.1.2 Základní pojmy

V této části budou vysvětleny základní pojmy a termíny týkající se hry sudoku. Všechny pojmy budou vysvětleny takovým způsobem, aby je pochopil i běžný laik, který tuto hru ještě nikdy předtím neviděl nebo nehrál.



Obrázek 2.1: Základní popis hracího pole sudoku

## Vymezení pojmů

*Hra sudoku* – je obecný název hry

*Buňka* – je nejmenší prvek (políčko), z kterých je vytvořeno celé hrací pole sudoku. Klasická hra sudoku má 81 buněk

*Díra* – prázdná buňka, která neobsahuje žádné číslo

*Box* – je čtverec, který má velikost 3x3 buňky, klasické pole sudoku má tedy 9 boxů

*Pole sudoku* – je obecné hrací pole hry sudoku (v našem případě 9x9 buněk)

*Mřížka sudoku* – je obecné pole sudoku bez dír, tedy slouží jako řešení hry

*Úloha sudoku* - je pole sudoku obsahující díry i vyplněná políčka, tedy jedná se o zadání hry

*Kandidát* – je číslo z rozsahu 1-9, které se doplňuje do jednotlivých buněk

9					4			1
	1	2						6
8	5	4	2	1				3
2			8				1	4
6	4				1	5	7	
	8			7		6	3	2
3		1		9		4	8	5
4	2	8	5		7	1		9
			1	4	8	3		

9	6	3	7	8	4	2	5	1
7	1	2	9	5	3	8	4	6
8	5	4	2	1	6	7	9	3
2	3	7	8	6	5	9	1	4
6	4	9	3	2	1	5	7	8
1	8	5	4	7	9	6	3	2
3	7	1	6	9	2	4	8	5
4	2	8	5	3	7	1	6	9
5	9	6	1	4	8	3	2	7

Obrázek 2.2: Úloha a mřížka sudoku

### 2.1.3 Koncepce a pravidla hry

Klasické sudoku má 81 buněk (políček) do kterých se vkládají jednotlivá čísla rozsahu 1 až 9. Tyto buňky tvoří čtverec (pole sudoku) o rozměru 9 x 9 buněk (obecně  $n^2 \times n^2$  buněk), který se dále dělí na devět boxů o rozměru 3 x 3 buňky (obecně  $n \times n$  buněk). Takto připravené pole sudoku má vyplněné některé buňky jednocifernými číslicemi rozsahu 1 – 9 a tvoří úlohu sudoku.

Cílem hry je v co nejkratším čase vyplnit všechny buňky zadané úlohy sudoku podle určitých pravidel. Do každého z devíti boxů, řádků a sloupců úlohy sudoku se musí doplnit jednotlivé číslice z rozsahu 1 - 9 tak, aby se každá číslice v daném boxu, řádku a sloupci nacházela právě jednou.

### 2.1.4 Jednoznačné řešení hry

Aby měla úloha sudoku jednoznačné (unikátní) řešení, musí mít zadaných minimálně 17 čísel. Sudoku s 16 zadanými čísly nejsou známy. V případě, že má úloha sudoku zadáno pouze 17 čísel ještě neznámá, že musí být nejtěžší. Existují také těžší sudoku úlohy, které mají zadáno více než 17 čísel.

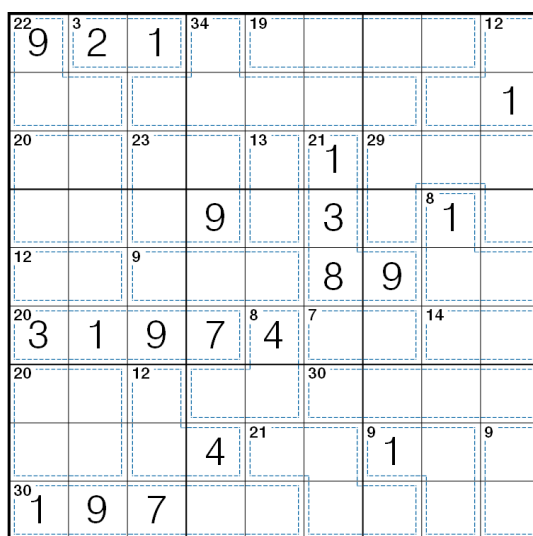
### 2.1.5 Celkový počet řešení

Do této doby nebyl objeven žádný obecný vzorec, který by umožnil vyčíslení počtu všech možných úloh sudoku pro obecný rozměr. V roce 2005 však Bertram Felgenhauer a Frazer Narcis vyčíslili počet všech možných úloh sudoku pro klasický rozměr 9 x 9 buněk, tento počet je asi 6 670 903 752 021 072 936 960, tedy  $6,67 \times 10^{21}$  tj. 6,67 triliardy [3].

### 2.1.6 Další varianty sudoku

Klasická hra sudoku, tak jak ji zná většina lidí je založená na hracím poli velikost 9 x 9 buněk do kterých jsou později doplněny jednotlivé číslice. Velikost hracího pole sudoku může být různá (existují varianty například 4 x 4 nebo 16 x 16 buněk). Tvar hracího pole nemusí být také čtverec, ale může to být například i koule (hra sudoku - ball).

Nejčastěji se do hracích polí doplňují číslíčky, ale jsou také varianty, kde je možno doplňovat písmena, tvary nebo různé obrazce. Samozřejmě, že pro tuto variantu platí stejné pravidla, kde v každém řádku a sloupci a boxu se písmeno nebo obrázek nachází právě jednou. Na základě těchto různých změn se začínají prosazovat i jiné druhy sudoku, které si získávají stále více fanoušků. Mezi tyto patří například diagonální sudoku, sudoku, kakuro, samurai a killer viz obrázek 2.3.



Obrázek 2.3: Ukázka killer sudoku

## 2.2 Požadavky na herní systém

Systém by měl být navržen tak, aby ho uživatelé mohli používat při každodenní činnosti. Měl by mít snadnou a jednoduchou navigaci v rámci celého systému. Zbytečně složité ovládání by mohlo uživatele odradit nebo případně hodně zneprjemnit jeho hru. Měl by podporovat přihlašování jednotlivých hráčů do systému (uživatelské jméno a heslo).

Herní server musí rychle a správně vyhodnocovat výsledky vyřešené hry sudoku. Tedy musí být dostatečně rychlá zpětná vazba mezi herním systémem a jeho uživatelem. Sudoku úlohy by měly mít různé úrovně obtížnosti, za účelem pokrytí co největšího spektra uživatelů.

## **2.3 Základní rozdělení problematiky**

Tuto práci je možno logicky rozdělit do dvou částí. První se zabývá samotným generátorem hry sudoku a druhá část řeší herní systém, na kterém bude tento generátor umístěn. Samotná problematika těchto dvou částí této práce, bude podrobně probrána v následujících kapitolách.

## 3 Generátor úloh sudoku

Cílem této práce bylo vytvořit generátor úloh sudoku, který je možno řešit třemi způsoby. V této kapitole budou popsány jednotlivé způsoby generování úloh sudoku a další metody, které se používají při generování.

### 3.1 Použití hotových úloh sudoku

Tento princip generování je velice triviální. K dispozici máme určitou úlohu sudoku s určitým počtem děr a obtížností. Jednoduchost tohoto algoritmu spočívá v záměně řádků nebo sloupců. Tato záměna musí být provedena tak, aby nebyla porušena pravidla sudoku.

V případě klasického sudoku, kde počet řádků a sloupců je roven devíti, mohou být zaměněny pouze řádky nebo sloupce v rozmezí velikosti každého boxu (v tomto případě velikostí tři). Tento algoritmus je velice rychlý, ale má omezený počet vygenerovaných úloh sudoku. Mohl by se využít v kombinaci s jinými generátory.

### 3.2 Odebírání z mřížky sudoku

Tento generátor pracuje s mřížkou sudoku (má vyplněny všechny buňky). Náhodně odebere číslo s mřížky sudoku (vytvoří díru) a provede kontrolu, zda má sudoku úloha jednoznačné řešení. Ke kontrole se používají řešící metody sudoku, které se snaží vyřešit takto vzniklou sudoku úlohu. Tyto metody řešení sudoku jsou popsány v této kapitole.

Pokud má sudoku úloha po odebrání čísla pouze jedno řešení, pokračuje generátor dalším odebráním čísla s mřížky sudoku. V případě, že po odebrání čísla bude mít sudoku úloha více než jedno řešení, dochází k porušení pravidel hry a toto číslo nesmí být s mřížky sudoku odebráno.

Proces generování sudoku úlohy se opakuje tak dlouho, dokud není z mřížky sudoku odebráno přesně  $N$  zadaných čísel ( $N > 17$ , aby bylo dodrženo jednoznačné řešení hry), kdy musí platit, že nově vzniklá úloha sudoku má pouze jedno řešení.

### 3.2.1 Generátor mřížky sudoku

Generátor, který pracuje na principu odebírání čísel z mřížky sudoku, musí mít tuto mřížku vyplněnou tak, aby neporušovala pravidla hry. V této podkapitole bude popsáno jakým způsobem je možno danou mřížku vytvořit.

Při vytváření mřížky se začíná s prázdným polem sudoku, které je rozdělené na jednotlivé boxy (čtverce  $n \times n$  buněk). Náhodně je vybrán jeden z boxů tohoto pole, kde se náhodně vyplňují čísla do jednotlivých buněk. Při vyplnění každé buňky musí být provedena kontrola, zda nedošlo k porušení pravidel hry (pole sudoku musí mít v každém řádku, sloupci a boxu číslo ze zvoleného intervalu obsaženo právě jednou). Pokud k takovému porušení pravidel dojde, je nutno se vrátit o jeden nebo více kroků zpět a vygenerovat celý box znovu.

Celý cyklus generátoru se opakuje tak dlouho, dokud nejsou naplněny všechny boxy pole sudoku. Takto vygenerované boxy v hracím poli tvoří požadovanou mřížku sudoku, která je nutná pro vytváření sudoku úloh.

### 3.3 Přidávání do prázdného pole sudoku

Tento generátor začíná pracovat s prázdným polem sudoku a funguje opačným způsobem jako generátor, který postupně odebírá čísla s mřížky sudoku. Do prázdného pole sudoku se postupně doplňují čísla do náhodně zvolených buněk. Při doplnění čísla do buňky, dochází ke kontrole jedinečnosti této sudoku úlohy. Ke kontrole se použijí řešící metody sudoku, které se snaží zadanou úlohu vyřešit. Generování končí přidáním zadaných  $N$  čísel ( $N > 17$ ) do pole sudoku v případě, že nově vzniklá úloha sudoku má pouze jedno řešení.



## 3.4 Metody řešící sudoku úlohy

### 3.4.1 Metoda Backtracking

Nejedná se přímo o metodu řešení sudoku úlohy, ale o metodu prohledávání s návratem, používanou při řešení algoritmických úloh založenou na metodě „pokus – omyl“. Backtracking je tedy obecná metoda, která prohledává stavový prostor řešení úlohy do hloubky. Dokud je to možné snaží se prohledávat stavový prostor řešení z počátečního stavu směrem k cílovému (koncovému) stavu. Pokud je další postup nemožný, vrací se o krok (někdy několik kroků) zpět a hledá další cesty k cílovému řešení.

#### Ukázka základního pseudokódu metody backtracking

```
bool BackTrack(konfigurace)
{
    if (není jiná možnost)
        return (konfigurace je v cílovém stavu);

    for (pro všechny možnosti)
    {
        výběr jedné možnosti M;
        if (BackTrack(proved možnost M v konfiguraci)) return true; // rekurze
        zruš možnost M;
    }
    return false; //po vyzkoušení všech možnosti nenašel řešení
}
```

Pomocí této metody je možno řešit také zadanou sudoku úlohu nebo dokonce vyplnit prázdnou mřížku sudoku. Více informací o tomto řešení bude popsáno v následující kapitole rekurzivních algoritmů 3.4.2.

### 3.4.2 Rekurzivní algoritmus Brute-force

Jedná se o klasický rekurzivní algoritmus, který při řešení využívá hrubou sílu. Nejprve prochází jednotlivé buňky sudoku úlohy (zadání sudoku). V místech, kde jsou díry (chybí čísla) si určí jednotlivé kandidáty na dané pozice. Jednotliví kandidáti jsou určeni tak, aby neporušovali základní pravidla sudoku. V každém řádku, sloupci a boxu se může každá číslice ze zvoleného intervalu nacházet právě jednou. Dále hledá takové buňky, které mají nejmenší počet kandidátů, pokud je pro danou buňku pouze jeden kandidát, je odebrán z možných kandidátů a přidán jako řešení na stálou pozici v úloze sudoku [4].

Jelikož pro některé buňky existuje více kandidátů je potřeba použít řešení pomocí metody backtracking. V buňce, kde je nejmenší počet kandidátů se vybere jeden a vloží na stálou pozici a dojde ke změně hracího pole a musí se přepočítat všichni kandidáti dané hry. Pokud se mu tímto způsobem podaří vyplnit všechny buňky je vše v pořádku. V opačném případě, kdy jsou vyčerpány všichni kandidáti pro danou buňku je potřeba se vrátit o krok zpět a vybrat jiného kandidáta u předchozí buňky.

Výhodou toho algoritmu je jeho malá délka kódu, který lze napsat na pár málo řádků a jeho obecnost, kdy tento algoritmus je možno použít na sudoku úlohu obecné velikosti  $n$ . Nevýhodou této metody je, že dokáže být velice pomalá ve srovnání s metodami napodobňující lidské logické myšlení. Existují sudoku úlohy, které vyřeší velice rychle, ale také úlohy, které mohou trvat i desítky minut. Na obrázku 3.1 je sudoku úloha, která má první řádek prázdný a zadáno pouze 17 čísel, což je minimální počet pro jednoznačné řešení. Při použití algoritmu hrubé síly by proběhlo obrovské množství iterací a tomu odpovídající čas, a proto je na takové řešení lepší použít jiných metod, které budou popsány v následující kapitole 3.4.3.

					3		8	5
		1		2				
			5		7			
		4				1		
	9							
5							7	3
		2		1				
				4				9

Obrázek 3.1: Náročná sudoku úloha pro řešení algoritmu Brute-force

### 3.4.3 Techniky napodobující logické myšlení řešitele

#### Technika Full House

V zadané sudoku úloze na obrázku 3.2 je potřeba doplnit poslední číslici v pátém řádku do šedého políčka. V tomto případě zbývá doplnit číslici osm, což je jednoduše odvozeno. Tato situace se v sudoku nazývá „Last digits“, neboli poslední číslice.

Po doplnění poslední číslice dochází také k eliminaci doplněného kandidáta z blízkých buněk. V podstatě se nejedná přímo o techniku, ale pouze pravidlo, které se bude používat nejčastěji v poslední fázi při každém řešení [5].

		1			3			8
			5			9		3
				2	9			
	8					6		9
2	7	9	1	5	6		3	4
4		6					7	
			2	7				
3		2			1			
6			3				9	

Obrázek 3.2: Ukázka techniky Full House

#### Technika Cross hatching

Tato technika spočívá v doplnění kandidáta do boxu (v boxu může být číslo z intervalu 1-9 právě jednou). Při kontrole jednotlivých řádků nebo sloupců, nastává situace, kdy dochází k postupnému vyloučení kandidáta z okolních boxů, a proto mu zbývá pouze jedna možnost [5].

Na obrázku 3.3 je simulace této techniky, kdy do zvýrazněné buňky je možno doplnit pouze číslo 6. V ostatních sloupcích nelze číslo 6 doplnit, protože by došlo k porušení pravidel sudoku.

<b>6</b>			2	7				
3		2			1			
			3				9	
	8					6		9
2	7	9	1	5	<b>6</b>		3	4
4		<b>6</b>					7	
				2	9			
	2		5	1		9		3
	1				3			8

Obrázek 3.3: Ukázka techniky Cross hatching 1

Na obrázku 3.4 je další simulace této techniky, kdy do zvýrazněné buňky je možno doplnit pouze číslo 8. V ostatních případech nelze číslo 8 doplnit, protože by došlo k porušení pravidel sudoku.

		<b>6</b>						
		4	2			5		<b>8</b>
7			<b>8</b>					
9		1					2	
			9	4				
							1	
5				8		2	9	
	6				2	1		
<b>8</b>		9		1	6		5	7

Obrázek 3.4: Ukázka techniky Cross hatching 2

## 4 Herní systém

Druhým cílem této práce je vytvoření webového herního systému pro zájemce o hru sudoku. Herní server bude uživateli nabízet různé soutěže turnaje, kde si bude moci otestovat své hráčské schopnosti. Také bude místem pro odreagování a relaxování jednotlivých uživatelů. Součástí nebudou pouze soutěžní hry, ale také možnost hrát různé tréninkové hry.

V této kapitole budou popsány jednotlivé části tohoto systému, které lépe přiblíží jeho fungování. Budou popsány jednotlivé uživatelské role v systému a další důležité informace týkající se soutěží, her, herních týmů, číselníků a výsledků, které bude tento systém nabízet.

### 4.1 Role v systému

V systému jsou definováni jednotliví aktéři, kteří s ním komunikují. Nejedná se přímo o uživatele, ale o jejich role. Jeden uživatel může v rámci systému vystupovat jako několik aktérů, jinak řečeno ve více rolích. Aktéři se mohou odlišovat svými právy (technický administrátor má větší pravomoci než hráč) nebo způsobem zacházení s daty (hráč může zakládat nové týmy, kdežto anonym pouze číst).

#### 4.1.1 Anonym

Neregistrovaný uživatel, který má volný přístup na herní portál. Na serveru si může přečíst všechny aktuality a novinky o soutěžích, které momentálně probíhají. Má možnost volně trénovat hru sudoku ve zvolené obtížnosti, kde se mu po vyřešení hry zobrazí čas jeho řešení. Může si také prostudovat pravidla a strategii hry, které budou součástí herního portálu v sekci nápověda. Prohlédnout si může také jednotlivé hráčské týmy. V případě velkého zájmu o tuto hru se může zaregistrovat jako hráč a účastnit se různých soutěží, což je více popsáno v kapitole 4.2.

### **4.1.2 Hráč**

Registrovaný uživatel, který má své konto v rámci herního serveru. Má stejné všechny práva a přístup k datům jako anonym a k tomu něco navíc. Účastní se různých soutěží, kde při úspěšném vyřešení hry může získat určitý počet bodů. Tyto body budou závislé na rychlosti vyřešení dané hry a budou se hráči v rámci soutěže a konta postupně sčítat. V rámci svého konta bude moci sledovat svou úspěšnost (podle času řešení, počtu bodů, vyřešených her). Hráč může založit nebo se stát členem, již založeného hráčského týmu, který bude reprezentovat v rámci soutěží.

### **4.1.3 Administrátor**

Role administrátora se může rozdělit na dva typy:

- Technický administrátor (správce systému)
- Administrátor soutěží (správce soutěží)

#### **Technický administrátor**

Úkolem technického administrátora je celková správa systému, kde patří vytváření záloh a přidávání informací do systému. V systému bude také přidávat novinky o soutěžích a různé aktuální zprávy v sekci aktuality. Na starosti má aktualizaci jednotlivých číselníků a nasazování nových verzí systému. V případě objasnění podvodu při řešení nějaké hry v rámci soutěže, má možnost danému hráči odepřít přístup na herní server a to odebráním z evidence uživatelů (hráčů).

#### **Administrátor soutěží**

Administrátor soutěží má na starosti veškerou správu soutěží. Správa soutěží je dána založením soutěže, kde administrátor vymyslí vhodný název soutěže, termín, interval konání a zvolí odpovídající obtížnost generovaných her. Soutěže může podle potřeby upravovat nebo také mazat. Mazání je možné pouze pokud v dané soutěži ještě nehrál ani jeden hráč.

Dále má možnost diskvalifikovat hráče v soutěži na základě prokázaného podvodu hráče ve hře. V případě, opakovaného podvodu, bude hráči odepřen přístup na herní server, o který se postará technický administrátor, zrušením jeho účtu na serveru.

## 4.2 Hráč

Hráčem je každý uživatel, který má přístup do herního systému. Tento přístup je dán registračním procesem, kdy se budoucímu hráči vytvoří konto na serveru. Na své konto se uživatel dostane pomocí přihlašovacích údajů, které zadá při registraci.

Hráč se může účastnit zvolených soutěží, kde mu bude vygenerována příslušná úloha sudoku k řešení. Po rychlém a úspěšném vyřešení této hry budou hráči připsány body za hru. Své body a další údaje týkající se jeho soutěžení si bude moci prohlédnout na svém účtu v rámci herního serveru.

U jednotlivých hráčů se bude evidovat:

- jméno
- příjmení
- pohlaví
- uživatelské jméno
- heslo
- email
- pohlaví
- národnost
- věková skupina
- herní tým

Může se stát také členem nějakého herního týmu, který bude sdílet s více hráči. Tento tým bude reprezentovat při účasti na různých soutěžích. Pokud bude chtít, má možnost si založit také vlastní herní tým.

## 4.3 Soutěž

Herní server umožní registrovanému uživateli účast v soutěžích, které si sám vybere. Každá soutěž bude probíhat v předem určeném časovém období (bude mít stanoven začátek a konec soutěže). Bude mít definovanou obtížnost generovaných her (lehká, střední a těžká). V rámci soutěže budou ve vybraném intervalu (jeden, dva a více dnů, týden) a určeném čase vygenerovány všechny soutěžní hry (sudoku úlohy k řešení).

U jednotlivých soutěží se bude evidovat:

- název
- začátek soutěže
- konec soutěže
- časový interval
- čas generování hry
- popis

Veškerou správu a dění kolem soutěží má na starosti administrátor soutěží. Ten danou soutěž musí založit, vyplní vhodný název, popis, začátek a konec soutěže, vybere vhodný časový interval (př. každý den v osm hodin, každou neděli v deset hodin) a obtížnost pro vygenerování nových her.

## 4.4 Hra

Hrou se rozumí vygenerována sudoku úloha v rámci určité soutěže, kterou řeší registrovaný uživatel v roli hráče. Konkrétní hra je vygenerována v časovém intervalu, který je předem určen v soutěži.

Hráč si nejprve načte zadání hry, kdy se mu začne počítat čas řešení, a proto se pokouší hru vyřešit v co nejkratším čase. Po vyřešení odešle hru ke kontrole. Pokud je řešení správné, spočítají se body a stav hry se mění na vyřešenou. V případě špatného řešení je počet bodů nulový a stav hry se mění na nevyřešenou.



U jednotlivých her se bude evidovat:

- hráč hry
- sudoku úloha
- začátek hry
- konec hry
- počet bodů
- stav

#### **4.4.1 Bodování**

Každá hra bude mít podle obtížnosti hry přidělen určitý počet bodů (př. lehká 5000bodů, střední 7000bodů a těžká 10000bodů). Tyto body se odečítají v závislosti na výsledném času řešení konkrétní hry, kdy se za každou sekundu řešení odečte určitý počet bodů. Hru vyhraje uživatel, který má největší počet bodů, tedy vyřešil hru v nejlepším čase. Ovšem pokud bude hra vyřešena špatně, je počet bodů automaticky nulový.

### **4.5 Týmy**

Každý hráč může být členem některého z herních týmů sudoku. Tento tým může reprezentovat při různých soutěžích. Svůj tým si může založit každý s hráčů herního systému, kde může mít skupinu hráčů, kteří se s touto skupinou ztotožní a stanou se rovnoprávními členy.

U jednotlivých týmů se bude evidovat:

- název
- popis
- zakladatel

## 4.6 Číselníky

Nedílnou součástí systému jsou také číselníky, které má na starosti technický administrátor. U hráče to bude číselník národností a věkových skupin. V rámci soutěže to bude číselník obtížností, kde u každé obtížnosti bude maximální počet bodů za hru. Dále to bude číselník časových intervalů, který se bude používat v soutěžích pro generování her.

## 4.7 Průběžné a závěrečné výsledky

Výsledky jednotlivých hráčů dosažené při řešení různých soutěží je možno sledovat ze dvou pohledů. První může být brán z pohledu jednotlivého hráče, který sleduje vlastní výsledky v rámci svého profilu. Na druhé straně jsou to hromadné výsledky v rámci soutěží, které mohou sledovat všichni uživatelé herního systému.

V části uživatelského profilu každého hráče budou evidovány tyto výsledky:

- celkové pořadí v soutěžích
- celkový počet bodů
- celkový čas řešení
- počet úspěšně vyřešených her
- počet neúspěšně vyřešených her

V sekci výsledky v rámci systému budou generovány tyto výsledky:

- celkový výsledek všech soutěží za vybrané časové období
- celkový výsledek vybrané soutěže
- průběžný výsledek vybrané aktuálně probíhající soutěže
- průběžné výsledky aktuálního dne, měsíce, roku

U těchto výsledků budou vypisováni jednotliví hráči podle počtu získaných bodů (od vítěze, až po nejhoršího řešitele). V rámci těchto výsledků se budou vypisovat ještě další informace: pořadí v soutěži, uživatelské jméno hráče, jeho herní tým, počet bodů získaných v soutěži, počet odehraných her a výsledný čas řešení.

## 5 Analýza

Analýza je další fází softwarového procesu a pokrývá převod slovního zadání či jinak zpřesněného popisu reality do tvaru, který může sloužit jako podklad pro zahájení návrhu řešení a pro vlastní implementaci. Analýza znamená studium problému (jeho poznání, popis, modelování) dříve, než se začnou provádět akce směřující k řešení tohoto systému.

### 5.1 Datová analýza

Cílem datové analýzy je vytvořit model budoucího systému z pohledu datových struktur, se kterými bude systém pracovat. Jedná se o proces poznávání objektů reálného světa, jejich vlastností a vazeb, které jsou nedílnou součástí pro budoucí systém [6].

Datová analýza se skládá z těchto částí:

- lineární zápis typů entit
- lineární zápis typů vazeb
- E-R diagram
- popis a specifikace atributů
- datový slovník (entity, relace, atributy, atd.)

V této podkapitole jsou uvedeny pouze některé části datové analýzy, které se snaží přiblížit budoucí systém. Kompletní datová analýza je součástí obsahu CD, které je umístěno v příloze této práce.

#### 5.1.1 Lineární zápis

Lineární zápis obsahuje specifikace entitních typů a vztahů mezi těmito entitními typy, které jsou realizovány prostřednictvím klíčových atributů.

### Ukázka lineárního zápisu typů entit

**Hrac**(hrac\_id, hrac\_pohlavi, hrac\_jmeno, hrac\_prijmeni, hrac\_email, *narodnost\_id*, *vek\_skupina\_id*, *tym\_id*, *user\_name*, hrac\_aktivita)

**Hra**(hra\_id, *hrac\_id*, *sudoku\_id*, hra\_zacatek, hra\_konec, hra\_body, stav\_hry\_id)

**Sudoku**(sudoku\_id, *soutez\_id*, sudoku\_zadani, sudoku\_zacatek, sudoku\_konec, *res\_sudoku\_id*)

**Res\_sudoku**(res\_sudoku\_id, sudoku\_mrzka, datum\_vytvoreni, typ)

**Soutez**(soutez\_id, soutez\_nazev, soutez\_zacatek, soutez\_konec, cas\_interval\_id, cas\_hry\_id, soutez\_popis)

**Tym**(tym\_id, tym\_nazev, tym\_zakladatel, tym\_popis)

**Stav\_hry**(stav\_hry\_id, stav\_hry\_nazev)

**Obtiznost**(obtiznost\_id, obtiznost\_nazev, obtiznost\_body)

**Narodnost**(narod\_id, narod\_nazev)

**Vek\_skupina**(vek\_skupina\_id, vek\_skupina\_nazev)

**Cas\_interval**(cas\_interval\_id, cas\_interval\_nazev, cas\_interval\_hodnota)

**Cas\_hry**(cas\_hry\_id, cas\_hry\_hodnota)

**Legenda:** [primární klíč, *cizí klíč*]

### Ukázka lineárního zápisu typů vztahů

JE\_ČLENEM (Hrac, Tym) **N:1**

MÁ (Hrac, Narodnost) **N:1**

JE\_STARÝ (Hrac, Vek\_skupina) **N:1**

HRAJE (Hrac, Hra) **1:N**

VE\_STAVU (Hra, Stav\_hry) **N:1**

OBSAHUJE (Hra, Sudoku) **N:1**

ŘEŠÍ (Sudoku, Res\_sudoku) **1:1**

SOUČÁSTÍ (Sudoku, Soutez) **N:1**

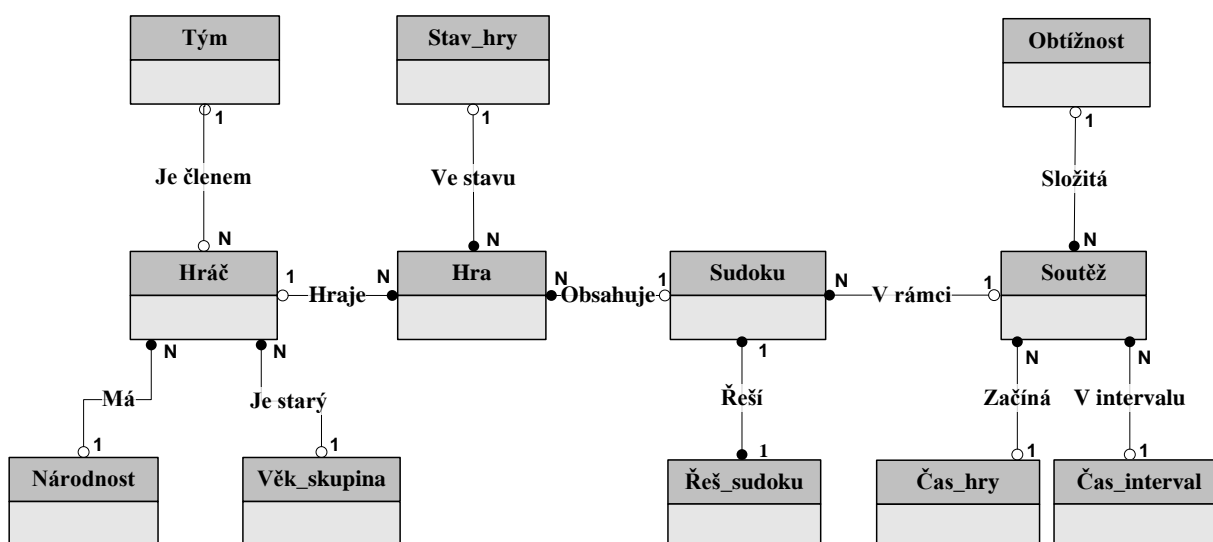
ZAČÍNÁ (Soutez, Cas\_hry) **N:1**

V\_INTERVALU (Soutez, Cas\_interval) **N:1**

SLOŽITÁ (Soutez, Obtiznost) **N:1**

### 5.1.2 E-R diagram

E-R model je nástroj pro detailní grafické a logické zachycení entit, jejich typů a vazeb. E-R diagram pomáhá na konceptuální úrovni abstrakce popsat uživatelskou aplikaci za účelem specifikovat následně strukturu databáze. E-R diagram herního serveru je zobrazen na obrázku 5.1



Obrázek 5.1: E-R diagram Herního serveru SUDOKU

### 5.1.3 Datový slovník

Slouží ke slovnímu formalizovanému popisu dat systému z pohledu uživatele.

#### Ukázka některých tabulek z datového slovníku této práce

<b>Sudoku</b> Popis: Pro evidenci sudoku							
Jméno atributu	Datový typ	Vel	Klíč	Nutno zadat	Cizí klíč	Integritní omezení	Popis
sudoku_id	int		A	A	N	číslo	autoincrement
soutez_id	int		N	A	A	číslo	id soutěže
sudoku_zadani	varchar	100	N	A	N	písmena	zadání hry
sudoku_zacatek	datetime		N	A	N		začátek generování
sudoku_konec	datetime		N	A	N		konec generování
res_sudoku_id	int		N	A	A	číslo	řešení sudoku id

<b>Soutěž</b> Popis: Pro evidenci soutěží							
Jméno atributu	Datový typ	Vel	Klíč	Nutno zadat	Cizí klíč	Integritní omezení	Popis
soutez_id	int		A	A	N	číslo	autoincrement
soutez_nazev	varchar	150	N	A	N	číslo	název soutěže
soutez_zacatek	datetime		N	A	N		začátek soutěže
soutez_konec	datetime		N	A	N		konec soutěže
soutez_cas	datetime		N	A	N		čas soutěže
soutez_interval	int		N	A	N	číslo	interval soutěže
soutez_popis	varchar	500	N	A	N	písmena	popis

<b>Hra</b> Popis: Pro evidenci her							
Jméno atributu	Datový typ	Vel	Klíč	Nutno zadat	Cizí klíč	Integritní omezení	Popis
hra_id	int		A	A	N	číslo	autoincrement
hrac_id	int		N	A	A	číslo	id hráče
sudoku_id	int		N	A	A	písmena	id sudoku
hra_zacatek	datetime		N	N	N		začátek hry
hra_konec	datetime		N	A	N		konec hry
hra_body	int		N	N	N	číslo	body hry
stav_hry_id	int		N	A	A	číslo	stav hry

<b>Hráč</b> Popis: Pro evidenci hráčů							
Jméno atributu	Datový typ	Vel	Klíč	Nutno zadat	Cizí klíč	Integritní omezení	Popis
hrac_id	int		A	A	N	číslo	autoincrement
hrac_pohlavi	char	1	N	A	N	písmena	pohlaví hráče
hrac_jmeno	varchar	50	N	A	N	písmena	jméno hráče
hrac_prijmeni	varchar	50	N	A	N	písmena	příjmení hráče
hrac_email	varchar	50	N	A	N	písmena , čísla	emailová adresa hráče
narodnost_id	int		N	A	A	číslo	národnost hráče
vek_skupina_id	int		N	A	A	číslo	věková skupina hráče
tym_id	int		N	N	A	číslo	tým hráče
hrac_login	varchar	50	N	A	A	písmena , čísla	přihlašovací login hráče
hrac_aktivita	bit	1	N	A	N	true/false	aktivita

<b>Res_sudoku</b> Popis: Pro evidenci řešení sudoku							
Jméno atributu	Datový typ	Vel	Klíč	Nutno zadat	Cizí klíč	Integritní omezení	Popis
res_sudoku_id	int		A	A	N	číslo	autoincrement
res_sudoku_mrizka	varchar	250	N	A	N	číslo	mřížka bez dír
res_datum_vytvoreni	datetime	100	N	A	N		datum vytvoření

<b>Obtížnost</b> Popis: Číselník pro popis obtížnosti ve hře							
Jméno atributu	Datový typ	Vel	Klíč	Nutno zadat	Cizí klíč	Integritní omezení	Popis
obtznost_id	int		A	A	N	číslo	autoincrement
stav_hry_nazev	varchar	100	N	A	N	písmena	název stavu hry
obtznost_body	int		N	A	N	číslo	maximální počet bodů

<b>Tým</b> Popis: Pro evidenci týmů							
Jméno atributu	Datový typ	Vel	Klíč	Nutno zadat	Cizí klíč	Integritní omezení	Popis
tym_id	int		A	A	N	číslo	autoincrement
tym_nazev	varchar	100	N	A	N	písmena	název týmu
tym_zakladatel	int		N	A	N	číslo	zakladatel týmu
tym_popis	varchar	500	N	N	N	písmena	popis

## 5.2 Funkční analýza

Na základě hotové datové analýzy je možno pokračovat funkční analýzou. Ta má za úkol popsat všechny operace, které je zapotřebí s daty v navržené databázi provádět. Obecně to je ukládání, modifikace a rušení dat, výpočty, třídění, vyhledávání informací, formátování výstupních informací apod. Funkční analýza opět vychází ze zadání, požadovaných vstupů, výstupů a funkcí [1].

V této podkapitole jsou uvedeny pouze některé části funkční analýzy. Více je uvedeno na CD, které je umístěno v příloze této práce.

Funkční analýza může pro tvorbu funkčního modelu využívat tyto nástroje:

- diagram datových toků (DFD)
- minispecifikace
- diagram aktivit
- datový slovník (datové toky, atd.)

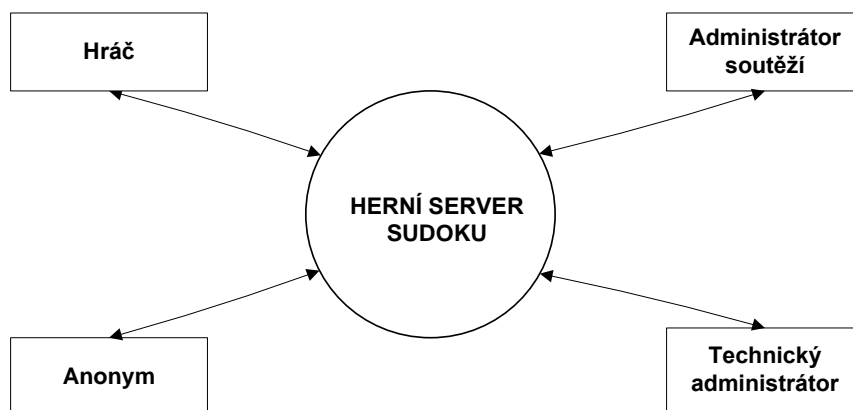
### 5.2.1 Diagramy datových toků (DFD)

Jedná se o první vnější pohled na funkce. Je to grafický prostředek pro návrh a zobrazení funkčního modelu systému. Podobně jako ERD u datové analýzy má být DFD dostatečně jednoduchý a názorný i pro uživatele a může sloužit i k upřesňování zadání [7].

#### Kontextový diagram

Obsahuje celý systém jako jednu funkci, definuje hranice systému a všechny aktéry.

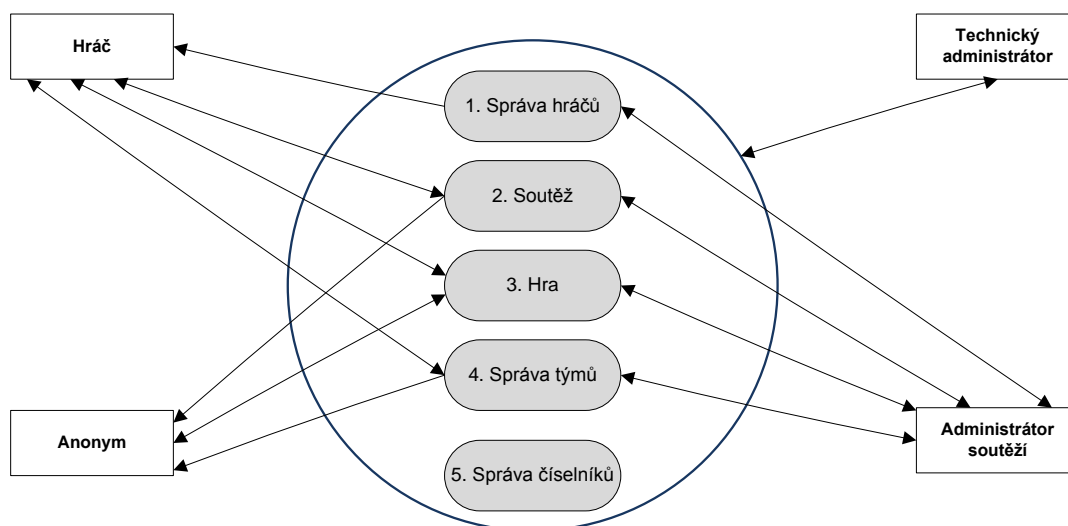




Obrázek 5.2: Kontextový diagram

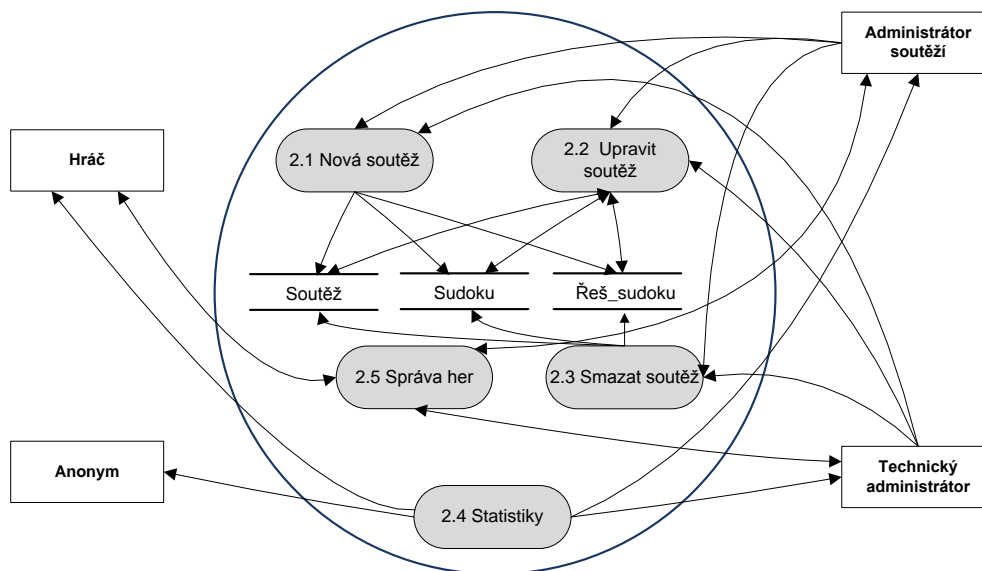
## DFD 0. úrovně systému

Tento diagram zobrazuje základní funkce systému a jejich vztahy. Tyto vztahy jsou popsány prostřednictvím paměti a datových toků. Aktéři tohoto systému jsou stejní jako u kontextového diagramu [7].



Obrázek 5.3: DFD 0. úrovně systému

## DFD 1. úrovně systému



Obrázek 5.4: DFD 1. úrovně systému

### 5.2.2 Minispecifikace

Minispecifikace jsou algoritmy elementárních funkcí pro nejnižší úroveň diagramu datových toků. V této části bude uvedena minispecifikace pro některý z procesů tohoto systému.

#### Řešení hry

- proces, který popisuje průběh řešení hry sudoku v rámci vybrané soutěže hráčem
- 1. ZOBRAZ formulář právě probíhajících soutěží
- 2. UŽIVATEL – HRÁČ vybere soutěž, kterou chce hrát
- 3. Podle vybrané soutěže DOPLŇ id soutěže do p.soutez\_id

4. NAČTI do p.datum aktuální datum
5. NAČTI záznam z tabulky sudoku do p.sudoku podle p.soutez\_id a p.datumu
6. DOPLŇ hodnoty z načteného záznamu do p.sudoku\_id, p.sudoku\_zadani a p.max\_bodu
7. NAČTI do p.zacatek\_hry aktuální datum a čas
8. NAČTI do p.hrac\_id id aktuálně přihlášeného hráče
9. PŘIDEJ nový záznam v tabulce hra podle proměnných p.hrac\_id, p.sudoku\_id, p.zacatek\_hry a p.status\_hry = nevyřešená
10. Po vložení záznamu v tabulce hra ULOŽ nové id hry do p.hra\_id
11. ZOBRAZ zadání hry podle p.sudoku\_hry
12. UŽIVATEL – HRÁČ vyplní a odešle hru
13. NAČTI do p.konec\_hry aktuální datum a čas
14. ULOŽ řešení hry do p.reseni\_hrac
15. ZJISTI id řešení sudoku do p.reseni\_sudoku\_id v tabulce sudoku podle p.sudoku\_id
16. NAČTI řešení dané hry do p.reseni\_hry z tabulky res\_sudoku podle p.reseni\_sudoku\_id
17. VYPOČTI celkový čas hry, kde  $p.cas\_hry = p.konec\_hry - p.zacatek\_hry$
18. VYNULUJ celkový počet bodů, kde  $p.pocet\_bodu = 0$
19. Jeli  $p.reseni\_hry \neq p.reseni\_hrac$  ZMĚŇ p.stav\_hry na nevyřešenou a BĚŽ na krok 21
20. SPOČTI celkový počet bodů do p.pocet\_bodu podle p.cas\_hry a p.max\_bodu
21. ZMĚŇ p.stav\_hry na vyřešenou
22. MODIFIKUJ záznam v tabulce hra, kde dojde ke změně atributů p.pocet\_bodu, p.konec\_hry a p.stav\_hry podle p.id\_hry
23. VYPIŠ informaci o úspěchu řešení dané hry

## 6 Návrh implementace

### 6.1 Architektura systému

Systém vytvořený na základě této diplomové práce bude postaven na třívrstvé architektuře s využitím technologie .NET. Třívrstvá architektura spočívá, v rozdělení aplikace na tři úrovně viz obrázek 6.1. Každá vrstva obsluhuje vlastní část celkového výpočetního procesu a sousedním vrstvám poskytne potřebné funkce. Jednotlivé vrstvy vždy komunikují pouze se sousední vrstvou modelů. Třívrstvý model nám umožňuje získat vyšší úroveň stability, jelikož provozní zátěž může běžet na více serverech.

#### Databázová vrstva

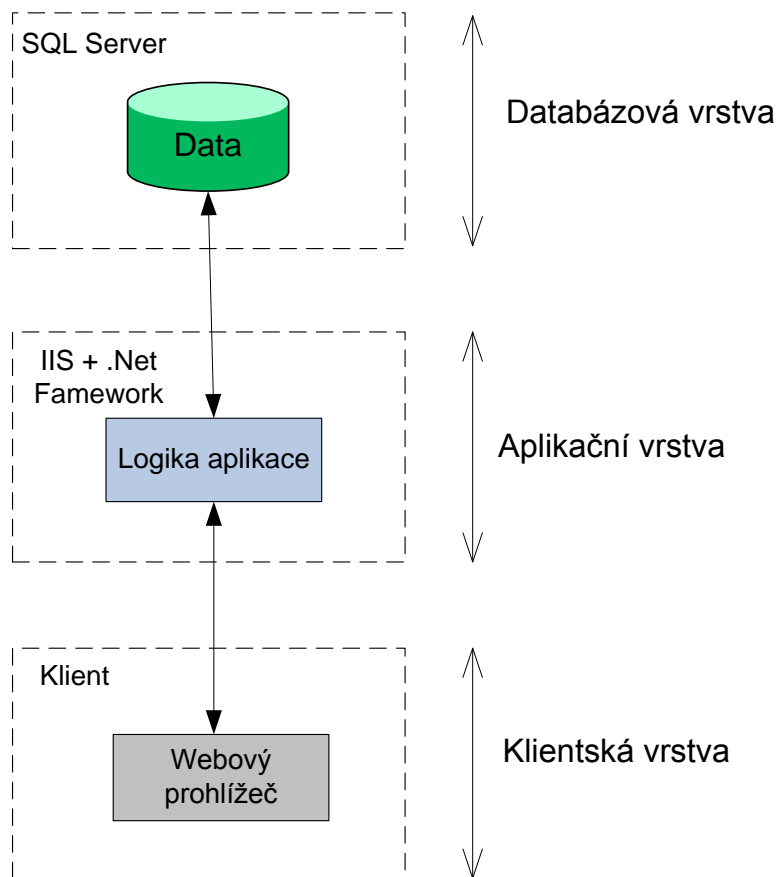
Nejnižší vrstvou je databázová vrstva, která zajišťuje práci se samotnými daty. Hlavním úkolem této vrstvy bude poskytovat spolehlivé úložiště dat pro tento systém a také zajišťovat jejich integritu. Dále tato vrstva bude poskytovat základní zabezpečení, rychlý přístup nebo vyhledávání dat. Všechny tyto operace bude zajišťovat databázový SQL server.

#### Aplikační vrstva

Střední vrstvou je aplikační vrstva, jež vytváří logiku celé aplikace. Jejím hlavním úkolem bude poskytovat všechny potřebné funkce klientské vrstvě. Aplikační server nemusí být nutně na jiném stroji než je server databázový, ale právě možnost vzájemného oddělení dává tomuto modelu výhodu velké rozšiřitelnosti. Jako aplikační server může být použit Internet Information Server (IIS).

#### Klientská vrstva

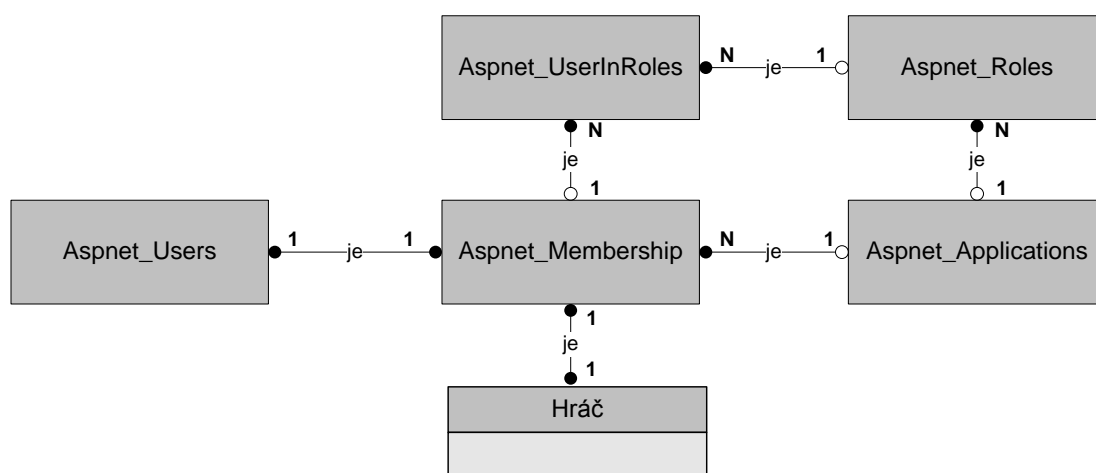
Poslední vrstvou je klientská vrstva, která poskytuje prezentační služby. Jejím dalším úkolem bude umožnit klientům využívat všech služeb celého systému, které definuje aplikační vrstva. Součástí této vrstvy budou webové formuláře a uživatelem definované komponenty.



Obrázek 6.1: Architektura herního serveru

## 6.2 Model autentizace a autorizace uživatele

Při přihlášení do systému bude muset uživatel zadat své jednoznačné přihlašovací jméno a heslo. Tyto údaje získá uživatel při registraci na herním serveru. Pro tuto autentizaci a pozdější autorizaci uživatele v systému bude použita komponenta založená na platformě .NET, jejichž konceptuální schéma je zobrazeno na obr. 6.2.



Obrázek 6.2: ER- diagram pro autorizaci a autentizaci uživatele prostředí .NET

### Krátký popis jednotlivých tabulek:

*AspNet\_Applications*- jednotlivé aplikace

*AspNet\_Users*- registrovaní uživatelé v rámci dané aplikace

*AspNet\_Membership*- obsahuje emaily a zašifrované hesla jednotlivých uživatelů

*AspNet\_Roles*- číselník všech možných rolí v aplikaci

*AspNet\_UserinRoles*- přiřazené role jednotlivým uživatelům

### Ukázka lineárního zápisu

**aspnet\_Applications** (ApplicationName, LoweredApplicationName, ApplicationId, Description)

**aspnet\_Users** (*ApplicationId*, UserId, UserName, LoweredUserName, MobileAlias, IsAnonymous, LastActivityDate)

**aspnet\_Membership** (*ApplicationId*, *UserId*, Password, PasswordFormat, PasswordSalt, MobilePIN, Email, LoweredEmail, PasswordQuestion, PasswordAnswer, IsApproved, IsLockedOut, CreateDate, LastLoginDate, LastPasswordChangedDate, LastLockoutDate, FailedPasswordAttemptCount, FailedPasswordAttemptWindowStart, FailedPasswordAnswerAttemptCount, FailedPasswordAnswerAttemptWindowStart, Comment, FirstName, Surname, Street, StreetNumber, City, DateOfBirth, Phone)

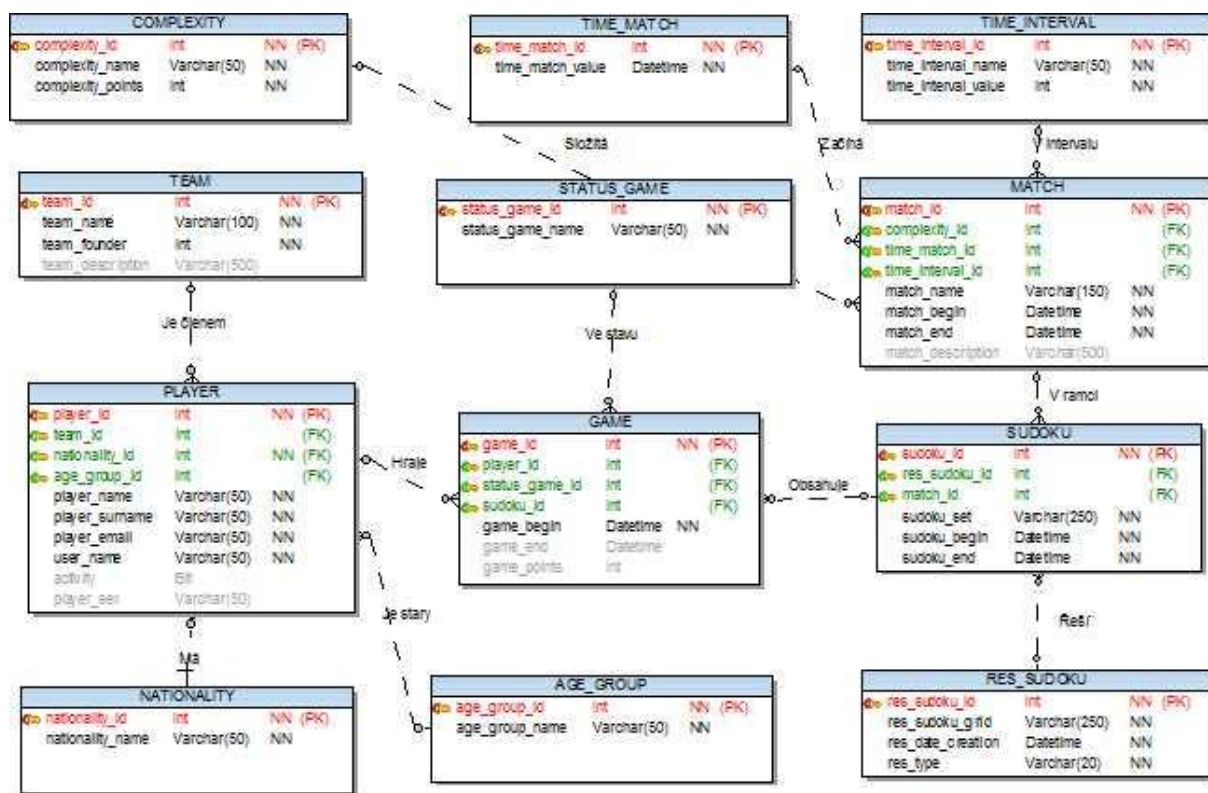
**aspnet\_Roles** (*ApplicationId*, RoleId, RoleName, LoweredRoleName, Description)

**aspnet\_UsersInRoles** (UserId, RoleId)

**Legenda:** [primární klíč, *cizí klíč*]

## 6.3 Model databáze

Na obrázku 6.3 je model databáze tohoto systému. Model databáze se snaží zobrazit jednotlivé objekty a vztahy mezi nimi s ohledem na budoucí implementaci. Jedná se o implementačně databázové schéma, které je možno implementovat v jakémkoli programovém prostředí.



Obrázek 6.3: Model databáze herního serveru

## 6.4 Transakční analýza

Cílem transakční analýzy je vyhledat části algoritmů v jednotlivých minispecifikacích, které reprezentují samotné transakce. Transakce je definována jako základní logická jednotka zpracovávání informací a musí proběhnout celá nebo musí být zajištěno její opakování [7]. Transakce je nutno řešit pokud se jedná o víceuživatelský systém, kde při práci s daty jednotlivých uživatelů, by mohlo dojít k nekonzistenci databáze.

Tento systém bude víceuživatelský, ale všichni aktéři v tomto systému budou mít přesně definované role a přístup k datům (editace, vkládání a mazání) bude mít v daném okamžiku vždy pouze jeden s uživatelů. Díky dobře rozděleným rolím v tomto systému, nebude docházet k vzájemnému přístupu více uživatelů ke stejným datům v jednom okamžiku. Nebude tedy docházet k nekonzistenci dat a není nutno provádět transakční analýzu.



## 6.5 Systémový návrh

### 6.5.1 ASP.NET

Tento systém bude vyvíjen pomocí technologie ASP.NET, což znamená: active server pages .NET.(aktivní serverové stránky.NET). Tato technologie je součástí .NET platformy firmy Microsoft pro tvorbu webových aplikací a služeb.

#### Popis technologie

Každá stránka v ASP.NET aplikaci obsahuje jednak HTML se serverovými komponentami, i kód v pozadí (tzv. code-behind), který ošetřuje události komponent a stará se o funkcionalitu celé stránky. Tento kód je vždy mimo HTML. Vzhled a obsah aplikace je tedy oddělen od aplikační logiky, díky čemuž je aplikace přehlednější. Část funkčnosti je možno zapsat přímo do značek serverových komponent, takže i poměrně složité stránky, které manipulují s daty, se dají napsat bez jediného řádku kódu. To je na druhou stranu kritizováno, protože to porušuje oddělení vzhledu od funkčnosti aplikace.

Celé ASP.NET je poskládáno z modulů, takže pokud se nám jakákoliv část ASP.NET nelíbí, je možno si ji poměrně snadno přizpůsobit. Ve většině případů stačí napsat nějakou třídu (s využitím dědičnosti) a použít ji místo vestavěného řešení, aniž by se tím ovlivnily ostatní části této platformy. Díky tomu je v ASP.NET možné opravdu vše. Je možno upravit prakticky každou funkcionalitu, kterou nám framework nabízí. Lze samozřejmě velmi jednoduše vytvářet vlastní komponenty skládáním několika jednodušších nebo úpravou těch stávajících.

#### .NET 3.5

Při vytváření herního systému bude použit .NET frameworku 3.5. Základní strukturu této platformy tvoří čtyři komponenty. První komponentou je базовá knihovna tříd (FCL), která obsahuje tisíce přednastavených tříd pro práci s datovými typy, kolekcemi, databázemi a mnoho dalších. Druhou komponentou je virtuální exekuční systém (CLR), který řídí a stará se o správné spouštění a běh aplikací .NET. Další komponentou je společný typový systém (CTS). Poslední komponentou je společná jazyková specifikace (CLS), která obsahuje základní sadu pravidel, které by měl splňovat každý programovací jazyk této platformy [8].

## **6.5.2 C#**

Jako programovací jazyk herního serveru bude použit jazyk C#, který vyvinula firma Microsoft a byl představen s vývojovým prostředím .NET. Návrh jazyka vychází ze zkušeností s jazykem C++, ale podobá se spíše programovacímu jazyku Java. Jedná se o čistě objektově orientovaný jazyk, který obsahuje nativní podporu komponentového programování. Dále zajišťuje typovou bezpečnost a podporuje řízení verzí a chyby dokáže zpracovat pomocí výjimek. Podporuje atributové programování a přidává vlastnosti a události k členským datům [10].

## **6.5.3 SQL Server**

Jako datové úložiště pro tento herní systém bude použit MS SQL server. Jedná se o relační databázový systém programovaný firmou Microsoft. Tato databáze bude využita také pro vytváření úložných procedur a funkcí, které budou užitečné při řešení logiky herního serveru.

### **Popis SQL serveru**

Jádro databázového serveru obsahuje databázové služby SQL Serveru, což představuje hlavní databázi (srdce tohoto serveru), replikace a komponenty pro fulltextové vyhledávání. Další částí jsou analytické služby, které poskytují služby pro obchodní systémy, umožňují agregovat data z více zdrojů. Integrované služby nabízejí podnikové transformace dat, kdy je dovoleno spojovat data z heterogenních datových zdrojů. Dále jsou to oznamovací služby, které zahrnují stroj na rozesílání oznámení a klientské komponenty pro včasné odesílání osobních zpráv uživatelům [11].

### **Nástroj pro práci**

Pro usnadnění práce s tímto databázovým serverem byl použit nástroj s názvem SQL Server Management Studio. Jedná se o integrované prostředí pro správu databázového serveru. Umožňuje velice jednoduše vytvářet databáze, tabulky, funkce nebo úložné procedury. Jeho součástí je i prostředí pro zadávání a ladění SQL příkazů, které je velice užitečné při vytváření různě složitých databázových příkazů.

## 6.5.4 AJAX

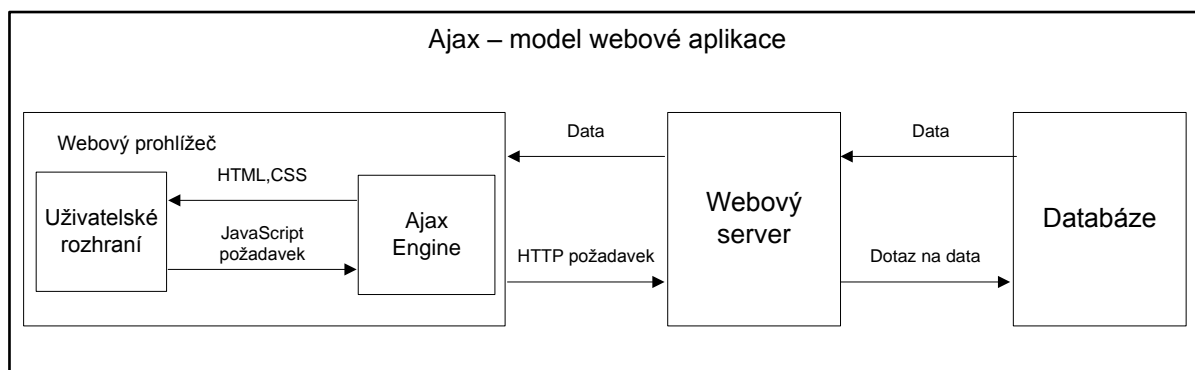
Ajax je zkratkou slov asynchronous javascript and xml a jedná se o moderní technologii často využívanou v současných webových aplikacích.

### Popis platformy

Ajax model (viz obrázek 6.1) poskytuje zprostředkující mezivrstvu nazvanou Ajax engine, která zajišťuje komunikaci mezi webovým prohlížečem a serverem. Ajax engine je většinou jen javascriptový objekt nebo nějaká funkce.

To je zásadní rozdíl od klasického přístupu, kde dojde k poskytnutí odkazu na další (externí) zdroj. V pojetí Ajaxu, každý odkaz na zdroj znamená volání Ajax engine. Požadavek je zpracován asynchronně, což znamená, že požadavek bude vyslán, ale nebude se čekat na jeho odpověď. Namísto toho se bude dále provádět kód klienta a odpověď bude zpracována později. Server, který obvykle dodává html dokumenty nebo javascriptový kód, bude nakonfigurován tak, že bude vracet data, která umí Ajax engine zpracovat. Tato data mohou být ve formě obvyčejného textu, xml dokumentů nebo v jiné formě. Jedinou podmínkou je, aby Ajax engine rozuměl a uměl interpretovat předávaná data. Když obdrží Ajax engine odpověď ze serveru, může provést příslušnou akci jako parsování dat, změnu podoby uživatelského rozhraní či poskytnutí informací uživateli.

Výhodou této technologie je nižší objem přenášených data a s tím související menší zátěž na webový server. Přináší rychlejší uživatelskou odezvu a vizuální přitažlivost Ajaxových komponent. Nevýhodou je použití velkého množství Ajaxových komponent v rámci webové stránky, kdy dochází k opačnému efektu zmíněných výhod.



Obrázek 6.4: Model webové aplikace AJAX

## 7 Implementace

Jedná se o vývojovou část všech předešlých etap, která zahrnuje vlastní programátorskou činnost. Výsledkem této činnosti je funkční herní server sudoku, který splňuje všechny požadavky, které byly uvedeny v kapitole analýzy a návrhu.

### 7.1 Úložné procedury a funkce

V rámci tohoto herního systému jsou některé databázové operace prováděny pomocí úložných procedur a funkcí v databázové vrstvě. Pomocí úložných procedur se v této práci řeší modifikace, vkládání nových záznamů viz obrázek 7.1 nebo různé dotazy nad databází viz obrázek 7.2. Pomocí funkcí jsou počítány některé informace týkající se hráče (celkové body, čas řešení, počet vyřešených her apod.) nebo prováděny kontroly viz obrázek 7.3.

```
CREATE PROCEDURE [dbo].[proc_MatchInsert]
(
    @match_name as varchar(150),
    @match_begin datetime,
    @match_end datetime,
    @time_match_id int,
    @time_interval_id int,
    @match_description as varchar(500),
    @complexity_id int,
    @new_id INT OUTPUT
) AS
BEGIN

    INSERT INTO MATCH(match_name, match_begin, match_end,
time_match_id, time_interval_id, match_description,
complexity_id) values (@match_name, @match_begin, @match_end,
@time_match_id, @time_interval_id, @match_description,
@complexity_id);

    SELECT @new_id = @@IDENTITY

END
```

Obrázek 7.1: Úložná procedura vloží nový záznam do tabulky soutěží a vrátí nové id soutěže.

```

CREATE PROCEDURE [dbo].[proc_staEvaluationPlayersByMatch]
(
    @match_id INT
) AS

SET NOCOUNT ON
DECLARE

    @poradi INT

BEGIN

SELECT
    ROW_NUMBER() OVER (ORDER BY Sum(game_points) DESC) AS poradi,
    player.user_name,
    team.team_name,
    Sum(game_points) as pocet_bodu,
    Count(*) as pocet_her,
    sum(datediff(second, game_begin, game_end)) as celkovy_cas

FROM MATCH
    JOIN SUDOKU ON sudoku.match_id = match.match_id
    JOIN GAME ON game.sudoku_id = sudoku.sudoku_id
    JOIN PLAYER ON player.player_id = game.player_id
    JOIN TEAM ON team.team_id = player.team_id

    WHERE match.match_id= @match_id

    GROUP BY player.user_name, team.team_name

END

```

*Obrázek 7.2: Úložná procedura zobrazí výsledky hráčů v rámci soutěže*

Tato úložná procedura zobrazí aktuální výsledky hráčů podle zadané soutěže. Jsou vypsány všichni hráči této soutěže podle počtu získaných bodů (pořadí od nejlepšího po nejhoršího). U každého hráče jsou kromě počtu získaných bodů za hru zobrazeny ještě tyto údaje: herní tým hráče, celkový čas řešení, počet odehraných her v rámci soutěže.

```

CREATE FUNCTION [dbo].[fn_ControlPlayer]
(
    @player_id as INT,
    @sudoku_id as INT
)
RETURNS BIT
AS
BEGIN
    DECLARE
        @count INT,
        @result BIT

    SET @result= 1
    SET @count= 0

    SELECT @count= Count(game_id)
        FROM GAME
        WHERE (sudoku_id= @sudoku_id) and (player_id =
            @player_id) and (game_end IS NOT NULL)

    IF (@count > 0)
        BEGIN
            SET @result= 0
        END

    RETURN (@result)
END

```

*Obrázek 7.3 Funkce kontroly hráče*

Funkce na základě id hráče a id sudoku provede kontrolu, zda uživatel danou hru ještě nehrál. Pokud je nalezen záznam s touto hrou a hráčem, vrátí funkce úspěch (1). V opačném případě, kdy není nalezen ani jeden záznam, tedy uživatel hru ještě nehrál a funkce vrátí neúspěch (0).

## 7.2 Použité komponenty

V této části budou popsány některé komponenty z prostředí ASP.NET a AJAX, které byly použity při implementaci této práce.

### 7.2.1 CalendarExtender

Jedná se Ajaxovou komponentu propracovaného kalendáře, kde slovo Extender znamená, že nefunguje samostatně, ale váže se na jinou komponentu v tomto systému (př. tlačítko, textBox).

```
<cc1:CalendarExtender
  ID="CalednarExtender"           // id komponenty
  runat="server"                  // běží na serveru
  Enabled="true"                  // povolit
  TargetControlID="TextBox"       // váže se na komponentu textbox
  PopupButtonID="Button"          // prvek, který zobrazí kalendář zde to je
                                  // kliknutí na tlačítko
  Format="dd.MM.yyyy"             // formát zobrazovaného datumu
  CssClass="kalendar"             // vzhled pomocí CSS
/>
</cc1:CalendarExtender>
```

Obrázek. 7.4: Ukázka kódu komponenty CalendarExtender



The image shows a web-based calendar control for May 2010. It features a header with navigation arrows and the month/year. The days of the week are abbreviated as Su, Mo, Tu, We, Th, Fr, Sa. The dates are displayed in a grid, with the current date (May 3, 2010) highlighted. Below the grid, the current date is displayed as 'Dnes: May 3, 2010'.

May, 2010						
Su	Mo	Tu	We	Th	Fr	Sa
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

Dnes: May 3, 2010

Obrázek 7.5: Vzhled komponenty CalendarExtender

## 7.2.2 TabContainer

Základní funkcí této Ajaxové komponenty je možnost přepínání mezi jednotlivými záložkami, které obsahují různé komponenty. Výhodou je, že při přepínání jednotlivých záložek se mění jejich obsah, ale nedochází k překreslení webového prohlížeče.

```
<ccl:TabContainer
  ID="tcStatistics"           // id komponenty
  runat="server"             // běží na serveru
>
  <ccl::TabPanel              // panel 1.záložky
    ID="tp1"                 // id 1.záložky
    runat="server"           // běží na serveru
    HeaderText="Denní výsledky" // nadpis záložky
  >
    <ContentTemplate>        // místo pro obsah

      // zde je obsah první záložky

    </ContentTemplate>
  </ccl::TabPanel>
</ccl:TabContainer>
```

Obrázek 7.8: Ukázka kódu komponenty TabContainer

Denní výsledky					
Měsíční výsledky					
Celkové výsledky					
Výsledky vybrané soutěže					
Pořadí	Uživatel	Člen týmu	Počet bodů	Počet her	Celkový čas
1	Honza	FC Brno	9073	1	0:05:09
2	Pavel	FC Ostrava	8995	1	0:05:35
3	Martin	FC Brno	8488	1	0:08:24

Obrázek 7.9: Vzhled komponenty TabContainer



## 7.2.3 GridView

GridView je mocný ovládací prvek z technologie ASP.NET, který si umí natáhnout datové řádky bez nutnosti použití velkého množství kódu. Tyto data se dají zobrazit v datové mřížce s možností výběru, editace, mazání, třídění, stránkování apod. Je vynikající k zobrazování kolekce údajů. Pomocí této komponenty se zobrazují všechny právě probíhající soutěže. Byl použit také v sekci výsledky, kde vypisuje různé průběžné i závěrečné hodnocení v rámci soutěží.

```
<asp:GridView
    ID="grvMatch"                // id komponenty
    runat="server"              // běží na serveru
    DataKeyNames="match_id"     // klíč dat
    DataSourceID="odsMatchList" // datový zdroj
    AutoGenerateColumns="False" // auto generování sloupců
    CssClass="vypis"           // třída stylů
    width="98%"                // velikost
>
    <Columns>
        // zobrazuje hodnotu pole v datovém zdroji
        <asp:BoundField />
        // zobrazuje příkazové tlačítko u každé položky
        <asp:ButtonField />
        // zobrazuje zaškrtačkové tlačítko u každé položky
        <asp:CheckBoxField />
        // tlačítka pro operace editace, vkládání, mazání apod.
        <asp:CommandField />
        // zobrazuje hodnotu dat. zdroje jako hypertextový odkaz
        <asp:HyperLinkField />
        <asp:TemplateField HeaderText="Název sloupce" >
            // umožňuje si vytvářet vlastní sloupce
        </asp:TemplateField>
    </Columns>
</asp:GridView>
```

Obrázek 7.6: Ukázka kódu komponenty GridView

Detail	Název hry	Začátek soutěže	Konec soutěže	Čas hry	Interval hry	Obtížnost	HRA
<a href="#">Zobraz</a>	Jarní sudoku	18.4.2010	18.6.2010	0:00	1 den	Lehká	<a href="#">Hraj</a>
<a href="#">Zobraz</a>	Velká cena Ostravy	27.4.2010	28.5.2010	20:00	1 den	Střední	<a href="#">Hraj</a>
<a href="#">Zobraz</a>	Brutal sudoku	1.5.2010	30.6.2010	8:00	1 den	Težká	<a href="#">Hraj</a>
<a href="#">Zobraz</a>	Květnová sudovka	1.5.2010	31.5.2010	20:00	1 den	Lehká	<a href="#">Hraj</a>

Obrázek 7.7: Ukázka vzhledu komponenty GridView

## 7.3 Hrací pole

Hrací pole sudoku se skládá s  $n \times n$  ( $9 \times 9$ ) textových políček (komponent typu TextBox). Toto pole je vykreslováno ve třech režimech, které mají různé nastavení vlastností jednotlivých políček v hracím poli.

### **PaintSudoku(int[,] grid)**

Vstupním parametrem je úloha (zadání) sudoku. Tato metoda vykreslí hrací pole s úlohou sudoku, kde jednotlivá čísla zadání jsou vykreslena modrou barvou a není možno je přepisovat.

### **PaintControlSudoku(int[,] gridPlayer, int[,] gridResolution)**

Vstupním parametrem je vyřešená úloha hráčem a správné řešení této úlohy. Metoda provede kontrolu řešení a v případě chybně vyplněného políčka, vykreslí toto políčko jinou barvou.

### **PaintSudokuResolution(int[,] gridPlayer, int[,] gridResolution)**

Vstupními parametry jsou hráčem vyřešená úloha a správné řešení této úlohy. Metoda provede kontrolu řešení a v případě chybně vyplněného políčka vykreslí toto políčko jinou barvou a doplní správné řešení.

4	9	3	8	6	2	5	1	7
5	7	2	3	4	1	9	8	6
1	8	6	7	5	9	4	2	3
3	2	1	9	7	5	8	6	4
9	6	7	4	1	8	5	2	7
8	4	5	6	2	3	1	9	3
9	2	9	9	8	5	2	1	9
2	6	3	1	3	6	5	4	8
7	1	8	2	9	4	6	5	3

Obrázek 7.10: Ukázka hracího pole při použití metody *PaintControlSudoku*

## 7.4 Obrazovka uživatelského rozhraní

Na obrázku 7.11 je obrazovka uživatelského rozhraní funkčního herního serveru sudoku. Každá obrazovka herního serveru je rozdělena do tří sekcí. Horní sekce obsahuje logo serveru a přihlašovací formulář pro uživatele. V sekci navigace jsou všechny ovládací tlačítka (Aktuality, Soutěže, Trénink, Týmy apod.), které umožňují rychlý a jednoduchý pohyb v rámci serveru. Součástí navigační sekce jsou také informace o přihlášeném uživateli. Poslední zobrazovací sekce má proměnlivý obsah v závislosti na použité navigaci a zobrazuje veškeré dění na herním serveru (v tomto případě obsahuje volné trénování hry sudoku).

Uživatelské jméno:  Heslo:   [Nová registrace](#) [Odhlásit](#)

Aktuality Soutěže Trénink Týmy Výsledky Nápověda Osobní profil Správa Serveru Přihlášený uživatel: Jarda

**Volné trénování hry SUDOKU**

Čas hry 00 : 24

	5		1				7	4
	2		9				6	
								2
					4		5	
		8				7		
4	3	6			9			
		3	7					8
		4		1	8			
						2		

Obtížnost:

Obrázek 7.11: Obrazovka uživatelského rozhraní herního systému

## 8 Testování

Nedílnou součástí vývoje webového systému je jeho dostatečné testování. Během celého vývoje tohoto systému bylo prováděno průběžné testování. Byly testovány, jak nově naimplementované části systému, tak i systém jako celek. Cílem bylo ověřit, že část testovaného systému (metoda, funkce, ovládací prvek aj.) dělá přesně to co má uvedeno v požadavcích. Testování bylo zaměřeno také na limitní a nestandardní situace, které dokážou analyzovat nejvíce chyb. V této kapitole budou uvedeny ukázky některých testovacích scénářů, které byly provedeny při vývoji herního serveru.

### 8.1 Testovací scénář přidání nové soutěže

<b>Název scénáře:</b>	Přidání nové soutěže
<b>Autor:</b>	Bc. Jaroslav Bližňák
<b>Datum vytvoření:</b>	5.3 2010
<b>Podmínky pro testování:</b>	Počítač s webovým prohlížečem Webový server s technologií .NET 3.5 Spuštěný databázový server
<b>Vstupy:</b>	Údaje pro založení nové soutěže
<b>Očekávané výstupy:</b>	Přidání nové soutěže na herním serveru
<b>Skutečné výstupy:</b>	Nová soutěž byla přidána, ale nezobrazila se ve výpisu soutěží

Tabulka 8.1: Přidání nové soutěže

Číslo	Popis	Výsledek
1	Spuštění URL adresy soutěží	v pořádku
2	Kontrola fungování tlačítka vytvoření nové soutěže	v pořádku
3	Kontrola zobrazených prvků pro přidání nové soutěže	v pořádku
4	Kontrola fungování prvků ve formuláři	v pořádku
5	Zadání údajů nové soutěže	v pořádku
7	Kontrola validace zadaných údajů	chyba
8	Kontrola funkce tlačítka stornovat novou soutěž	chyba
9	Kontrola funkce tlačítka vložit novou soutěž	v pořádku
10	Kontrola přidání nové soutěže ve výpisu soutěží	chyba

Tabulka 8.2: Průběh testování přidání nové soutěže

## 8.2 Testovací scénář registrace nového uživatele

<b>Název scénáře:</b>	Registrační proces
<b>Autor:</b>	Bc. Jaroslav Bližňák
<b>Datum vytvoření:</b>	15.3 2010
<b>Podmínky pro testování:</b>	Počítač s webovým prohlížečem Webový server s technologií .NET 3.5 Spuštěný databázový server
<b>Vstupy:</b>	Uživatelské registrační údaje
<b>Očekávané výstupy:</b>	Nový registrovaný uživatel v herním systému
<b>Skutečné výstupy:</b>	Registrace proběhla v pořádku, i přes některé nefunkční validace

*Tabulka 8.3 Registrační proces nového uživatele*

Číslo	Popis	Výsledek
1	Spuštění URL adresy registrace	v pořádku
2	Kontrola zobrazení registračního formuláře	v pořádku
3	Kontrola zobrazených prvků v prvním kroku registrace	v pořádku
4	Kontrola fungování prvků formuláře v prvním kroku	chyba
5	Zadání registračních údajů hráče	v pořádku
6	Kontrola validace zadaných údajů	chyba
7	Kontrola funkce tlačítka dalšího kroku registrace	v pořádku
8	Kontrola zobrazených prvků v druhém kroku registrace	v pořádku
9	Kontrola povinně zadaných údajů	chyba
10	Zadání dalších registračních údajů hráče	s chybou
11	Kontrola validace zadaných údajů	chyba
12	Kontrola zadání existujícího uživatele	v pořádku
13	Kontrola funkce tlačítka předchozího kroku	chyba
14	Kontrola funkce tlačítka provedení registrace	v pořádku
15	Kontrola zobrazených prvků ve finálním kroku registrace	v pořádku
16	Kontrola fungování tlačítka hotové registrace	v pořádku
17	Kontrola zda byla zobrazena úvodní stránka	v pořádku

*Tabulka 8.4 Průběh testování registrace nového uživatele*

Tento test proběhl s různými chybami. V prvním případě se jednalo o chybu nefunkčního rolovacího seznamu herních týmů. V druhém případě se jednalo o chyby validací některých povinných registračních položek. Všechny tyto chyby byly opraveny a znovu otestovány.

## 9 Závěr

Cílem této práce bylo vytvoření webového herního portálu pro zájemce o hru sudoku. Tento portál měl generovat úlohy sudoku různých obtížností. Součástí herního serveru měli být také různé soutěže v sudoku. V rámci hodnocení těchto soutěží se měli vypisovat různé průběžné i závěrečné výsledky. Bylo nutné nastudovat pravidla i strategie řešení hry a také podmínky pro generování nových zadání sudoku, které pro mě do této chvíle byly naprosto neznámé.

Výsledkem této práce je funkční herní server sudoku, který splňuje požadavky a cíle zadání. Podařilo se mi vytvořit uživatelsky jednoduché a příjemné prostředí tohoto serveru s generátorem jednoznačných her sudoku.

Při řešení této diplomové práce jsem si vylepšil své dovednosti s technologií ASP.NET 3.5. Zdokonalil jsem své znalosti s databázovým serverem MS SQL 2005, při použití různých úložných procedur a funkcí, které jsem použil v databázové vrstvě této práce. Rozšířil jsem také své vědomosti v rámci tvorby webových aplikací. Všechny tyto nově získané zkušenosti se pokusím využít v budoucí praxi.

V současné době je na naší internetové scéně velice málo online herních serverů sudoku. Doufám, že tento systém bude nasazen co nejdříve a najde si svou skupinu herních nadšenců. Vývoj systému bude probíhat i nadále a bude doplněn o další moduly. Dalším plánovaným modulem bude například vyhlašování různých soubojů v sudoku mezi jednotlivými hráči a modul diskusního fóra. Také hodnocení výsledků by mohlo být doplněno o různé vizuální grafy úspěšnosti. Aplikace by mohla být také upravena a ve zjednodušené formě použita na mobilních zařízeních.

## 10 Literatura

- [1] *Sudoku-hra* [online]. 2008 [cit. 2010-04-11]. Historie-sudoku. Dostupné z WWW: <<http://www.sudoku-hra.cz/>>.
- [2] *Online-sudoku* [online]. 2008 [cit. 2010-04-11]. Dostupné z WWW: <<http://www.online-sudoku.cz/>>.
- [3] *Mathematics of Sudoku* [online]. St. Petersburg (Florida): Wikipedia Foundation, 15.9.2005, 4.4 2010 [cit. 2010-04-19]. Dostupné z WWW: <[http://en.wikipedia.org/wiki/Mathematics\\_of\\_Sudoku](http://en.wikipedia.org/wiki/Mathematics_of_Sudoku)>.
- [4] *Algorithmics of sudoku* [online]. St. Petersburg (Florida): Wikipedia Foundation, 15.12.2006, 19.4.2010 [cit. 2010-04-19]. Dostupné z WWW: <[http://en.wikipedia.org/wiki/Algorithmics\\_of\\_sudoku](http://en.wikipedia.org/wiki/Algorithmics_of_sudoku)>.
- [5] *SudoCue* [online]. 2005 [cit. 2010-04-19]. Sudoku Solving Guide. Dostupné z WWW: <<http://www.sudocue.net/>>.
- [6] ŠARMANOVÁ, Jana. *Teorie zpracování dat* [online]. druhé přepracované. Ostrava: VŠB-Technická univerzita Ostrava, 2007 [cit. 2010-04-19]. Dostupné z WWW: <<http://www.elearn.vsb.cz/>>. ISBN 978-80-248-1498-8.
- [7] ŠARMANOVÁ, Jana. *Databázové a informační systémy* [online]. první. Ostrava: VŠB-Technická univerzita Ostrava, 2007 [cit. 2010-04-19]. Dostupné z WWW: <<http://www.elearn.vsb.cz/>>. ISBN 978-80-248-1499-5.
- [8] *C# 3.0: Programování na platformě .NET 3.5*. Brno: Zoner Press, 2009. 273 s. ISBN 978-80-483-5.
- [9] *ASP.NET* [online]. 2010 [cit. 2010-04-19]. Oficiální stránky ASP.NET. Dostupné z WWW: <<http://www.asp.net/>>.
- [10] SIMON, Robinson. *C# programujeme profesionálně*. Brno: Computer Press, 2003. 1105s. ISBN 80-251-0085-5.
- [11] WHALEN, Edward, et al. *Microsoft SQL Server 2005: Velký průvodce administrátora*. Brno: Computer Press, 2008. 1063s. ISBN 978-80-251-1949-5.
- [12] LACKO, Luboslav. *AJAX: Hotová řešení*. Brno: Computer Press, 2008. 263 s. ISBN 978-80-251-2108-5.

# 11 Přílohy diplomové práce

## 11.1 Obsah přiloženého CD-ROM

- \Analyza.pdf (analýza herního serveru sudoku)
- \TextPrace.pdf (text diplomové práce)
- \UzivatelaskaPrirucka.pdf (uživatelská příručka)
- \ProgramatorskaPrirucka.pdf (programátorská příručka)
- \ZdrojoveKody\Databaze (skripty pro vytvoření databáze)
- \ZdrojoveKody\HerniSystem (zdrojový kód herního serveru sudoku)